

融合 A* 与 DWA 算法的移动机器人动态避障研究*

鲁志 刘莹煌 张绪坤 侯睿

(南昌航空大学航空制造与机械工程学院 南昌 330063)

摘要: 针对传统 A* 算法在路径规划时存在搜索效率低、穿插障碍物、路径不平滑、无法规避未知障碍物等问题,提出了一种改进 A* 算法与改进 DWA 算法相融合的移动机器人动态避障方法。在改进 A* 算法中,引入全局障碍物占比率,在启发函数中增加动态权重系数,优化搜索领域,设定安全距离去除冗余节点,并加入三阶贝塞尔曲线对路径进行平滑处理,同时在 DWA 算法中加入目标点代价子函数,并动态调整代价函数系数,最后将改进 A* 算法与改进 DWA 算法进行融合,实现移动机器人的动态避障。仿真实验结果显示,在不同环境中,本文改进 A* 算法与传统 A* 算法以及其他改进 A* 算法相比,路径长度分别平均缩短了 5.14% 和 1.01%,搜索节点分别减少了 57.05% 和 36.59%,规划时间分别减少了 34.39% 和 8.49%;本文改进融合算法与传统融合算法以及其他融合算法相比,路径长度分别平均缩短了 19.89% 和 1.82%,规划时间分别平均减少了 53.66% 和 13.01%。证明了本文所提出的改进融合算法有效缩短了规划的路径长度与时间,能够在复杂的动态环境下实现实时避障,满足移动机器人行驶过程中的高效性和安全性。

关键词: A* 算法; DWA 算法; 移动机器人; 动态避障

中图分类号: TP242; TN102 文献标识码: A 国家标准学科分类代码: 510.8050

Research on mobile robot dynamic obstacle avoidance by fusing A* and DWA algorithms

Lu Zhi Liu Yinghuang Zhang Xukun Hou Rui

(College of Aerospace Manufacturing and Mechanical Engineering, Nanchang Hangkong University, Nanchang 330063, China)

Abstract: To solve the problems of low search efficiency, interspersed obstacles, unsmooth paths, and inability to avoid unknown obstacles in the traditional A* algorithm during path planning, a mobile robot dynamic obstacle avoidance method integrating the improved A* algorithm with the improved DWA algorithm is proposed. In the improved A* algorithm, the global obstacle occupancy ratio is introduced, dynamic weight coefficients are added to the heuristic function, the search field is optimized, the safe distance is set to remove redundant nodes, and third-order Bessel curves are added to smooth the paths, and the cost subfunctions of the target points are added to the DWA algorithm and the coefficients of the cost function are dynamically adjusted. The dynamic obstacle avoidance of the mobile robot is realized. Simulation results show that in different environments, compared with the traditional A* algorithm and other improved A* algorithms, the path length of the improved A* algorithm in this paper is shortened by an average of 5.14% and 1.01%, the search nodes are reduced by 57.05% and 36.59%, and the planning time is reduced by 34.39% and 8.49%, respectively; the improved fusion algorithm in this paper is reduced by an average of 5.14% and 1.01%, the search node is reduced by 57.05% and 36.59%, and the planning time is reduced by 34.39% and 8.49%, respectively. other fusion algorithms, the path length is shortened by 19.89% and 1.82% on average, and the planning time is reduced by 53.66% and 13.01% on average, respectively. It is proved that the improved fusion algorithm proposed in this paper effectively shortens the planned path length and time, and is able to realize real-time obstacle avoidance in complex dynamic environments to satisfy the high efficiency and safety of mobile robots during traveling.

Keywords: A* algorithm; DWA algorithm; mobile robot; dynamic obstacle avoidance

0 引言

近几十年来,机器人技术得到了迅速的发展,已被广泛

地应用于各个领域。其中移动机器人的路径规划是机器人技术中亟待深入探索与研究的关键方向之一,涉及到在给定的环境中确定机器人移动路径的问题^[1]。路径规划旨在

一定的环境中通过相应的算法找到一条从起始点到目标点的可行路径,且途中不会与任何障碍物发生碰撞^[2]。根据环境的不同,路径规划研究可分为静态全局规划和动态局部规划^[3]。全局路径规划是指在外界环境信息都已知的情况下,规划出一条最优路径。经典的全局路径规划算法主要有 Dijkstra 算法^[4]、A* 算法^[5]和快速扩展随机树算法 (rapidly-exploring random tree, RRT)^[6]等。局部路径规划是指在外界环境信息未知或部分已知的情况下,规划出一条无碰撞的路径,主要的算法有 D* 算法^[7]、人工势场法^[8]和动态窗口法 (dynamic window approach, DWA)^[9]等。

在全局路径规划中,A* 算法因其规划的路径短和较快的计算效率而被广泛应用。但在较为复杂的环境中,传统 A* 算法规划出来的路径存在搜索节点多、穿插障碍物、路径不平滑等不足,不利于移动机器人的运行。针对这些不足,众多学者对传统 A* 算法进行了多方面的优化与改进,并取得了显著的成效。唐嘉宁等^[10]提出一种双向搜索的 A* 算法,同时将双向搜索出来的路径进行了平滑处理,减少了路径中的冗余节点,提高了搜索效率。但这种方法在遇到复杂环境时规划出来的路径会从障碍物中间穿插过去,不利于移动机器人的安全行驶。张艳等^[11]通过改进邻域扩展方法,有效避免了经典四邻域扩展中存在的冗余节点多和八邻域扩展中存在的从障碍物中间穿过的路径等问题,同时设计了一种象限选择方法以及冗余点剔除策略,在路径搜索过程中可以有效减少多余节点的数量,但并没有考虑路径平滑问题。赖荣燊等^[12]将传统的八邻域搜索改为了五邻域搜索,减少了搜索节点,提高了计算效率,但并没有考虑优化后的 5 个搜索方向都存在障碍物的情况,从而陷入局部锁死问题。

以上方法只适用于障碍物已知的环境中,针对存在动态障碍物的环境,DWA 算法就表现出良好的有效性和鲁棒性。但是在环境比较复杂的情况下,DWA 算法容易陷入局部最优问题而导致路径规划失败^[13-15]。贺勇等^[16]通过引入折扣因子改进 DWA 算法的原始评价子函数,并提出新的速度平滑和轨迹相似度评价指标,有效地规避了未知障碍物。邓云峰等^[17]在 DWA 算法中引入碰撞锥对静、动障碍物进行避障检测,并通过速度窗口剔除掉具有碰撞威胁的速度来优化约束空间,提高了 DWA 算法的轨迹评估能力。罗征志等^[18]通过建立模糊逻辑系统自适应调整 DWA 算法评价函数的权重系数,实现了实时避障。贾千禧等^[19]通过引入人工势场法作为 DWA 算法的评价函数以选择最佳节点避免了与动态障碍物发生碰撞。然而,该算法仍存在躲避动态障碍物时易陷入局部最优的问题。

鉴于上述研究存在的问题,本文提出了一种改进 A* 算法和改进 DWA 算法相融合的移动机器人避障方法。在 A* 算法改进方面,引入全局障碍物占比率,对启发函数增加动态权重系数,并将搜索邻域从八邻域优化成六邻域,提高了路径搜索效率,同时根据设定的安全距离去除路径中

的冗余节点,并运用三阶贝塞尔曲线对路径平滑处理,减少路径长度;对于 DWA 算法的改进,在传统启发函数中加入目标点代价子函数,同时动态调整速度子函数的权重系数;最后将改进后 A* 算法和 DWA 算法进行融合,在优化后的全局最优路径上进行分段局部规划,实现移动机器人在动态环境下的自主避障。并通过仿真实验充分验证了所提出融合算法的有效性。

1 环境建模

在移动机器人路径规划中,常见的环境建模方法有可视图法、拓扑法、栅格法等^[20]。可视图法需要对环境信息进行详细的建模,在面对动态环境时,需要不断更新障碍物的位置和属性,增加了规划算法的复杂性。拓扑法对环境的建模主要是通过连接节点和边进行表示,当环境较为复杂时,节点和边的连接关系可能会变得非常复杂,导致规划算法的效率降低。而栅格法能够简化环境建模:将环境分割成网格后,可以将环境建模问题转化为一个图的搜索问题,每个单元格作为一个节点,无需考虑环境中的障碍物的具体形状,简化了路径规划的复杂度。由此本文选用栅格法进行环境建模。同时为了保证移动机器人行驶过程中的安全,根据实际障碍物的尺寸动态调整膨胀比例,当障碍物大小不足以占满一个栅格时,将其膨胀到一个栅格大小。

图 1 为 10×10 的栅格地图。其中黑色栅格为初始静态障碍物,红色栅格为新增静态障碍物,黄色栅格为动态障碍物,白色栅格为可通行区域。

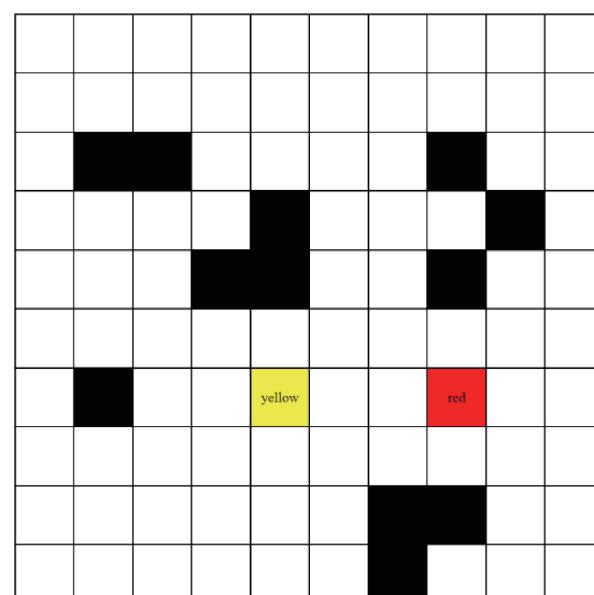


图 1 栅格地图

Fig. 1 Grid map

2 A* 算法及其改进

2.1 传统 A* 算法

A* 算法是一种启发式全局路径规划算法,在 Dijkstra

算法的基础上,融入广度优先搜索算法的优点演化而来^[21]。其原理是将当前节点视为父节点,通过搜索其周围的子节点,并计算每个子节点的代价值,选取最小代价值的子节点作为新的父节点,通过建立 Close 表记录已搜索过的节点,Open 表保存待扩展的子节点,同时引入启发函数来判断下一个节点的选择方向,重复这个过程直到找到目标点。

传统 A* 算法的评价函数表示为:

$$f(n) = g(n) + h(n) \quad (1)$$

其中, n 为当前节点, $f(n)$ 为起始点到目标点的总代价值; $g(n)$ 为起始点到当前节点 n 的实际代价值; $h(n)$ 为当前节点 n 到目标点的预估代价值。

常见的预估代价值计算方法有:

1) 欧几里德距离:

$$h(n) = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \quad (2)$$

2) 曼哈顿距离:

$$h(n) = |x_2 - x_1| + |y_2 - y_1| \quad (3)$$

3) 切比雪夫距离:

$$h(n) = \max\{|x_2 - x_1|, |y_2 - y_1|\} \quad (4)$$

其中, (x_1, y_1) 为当前节点的坐标, (x_2, y_2) 为地图点的坐标。本文选用欧几里德距离公式作为启发函数。

2.2 改进启发函数

在 A* 算法进行路径规划时,启发函数 $h(n)$ 常常发挥着重要作用。然而传统的启发函数的系数通常为常量,这使得在整个寻路过程中 $h(n)$ 的权值始终保持不变,降低了搜索效率。希望在当前节点距离目标点较远时,增加 $h(n)$ 的权重,缩小搜索空间,提高搜索效率,快速到达目标点;在当前节点离目标点较近时,减小权重,扩大搜索空间,使之安全到达目标点。

针对不同的地图环境,启发函数的权值也应有所不同。为此本研究引入了障碍物占比率 O ,抽象地描述了所处地图环境的复杂性。其定义为在移动机器人所处的整个地图区域内,障碍物栅格数与总栅格数的比值。公式如下:

$$O = \frac{N}{m \times n} \quad (5)$$

其中, N 为地图区域内障碍物栅格数, m 为地图区域的横坐标数, n 为地图区域的纵坐标数。

当移动机器人所处环境中障碍物较少时,增加 $h(n)$ 的权重,提高搜索效率;当障碍物较多时,减小权重,使之更加全面地探索所有可能的路径,避免陷入局部最优的问题。

由此提出了一种对启发函数动态加权的改进方法,其计算公式如下:

$$\omega = e^{\frac{d_1}{d_2}} - \frac{1}{2}(1 - e^{-O}) \quad (6)$$

$$d_1 = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \quad (7)$$

$$d_2 = \sqrt{(x_2 - x_0)^2 + (y_2 - y_0)^2} \quad (8)$$

$$f(n) = g(n) + \left[e^{\frac{d_1}{d_2}} - \frac{1}{2}(1 - e^{-O}) \right] h(n) \quad (9)$$

其中, (x_0, y_0) 为起始点的坐标, d_1 为当前节点到目标点的距离, d_2 为起始点到目标点的距离。

通过添加动态权重系数 ω ,可以更好地平衡启发函数在路径规划时的权重,解决了传统 A* 算法搜索效率低和局部最优问题。

2.3 优化搜索邻域

传统 A* 算法在进行路径规划时,通常会向当前节点 N 的周围 8 个领域子节点进行搜索,如图 2 所示。当目标点方位确定时,八邻域搜索会存在无用的栅格,造成运算时间和存储空间的浪费^[22]。因此提出了一种搜索邻域优化策略,根据目标点与当前节点的相对位置,减少为六邻域搜索。只有当周围六邻域都存在障碍物时,算法继续探索被舍弃的两个邻域。将当前节点与目标点的连线方向与正东方向的夹角设为 α ,对应的搜索邻域如表 1 所示。例如,当夹角 α 处于 $[0^\circ, 45^\circ]$,保留 1、2、3、4、7 和 8 节点为搜索邻域,舍弃 5 和 6 节点,如图 3 所示。

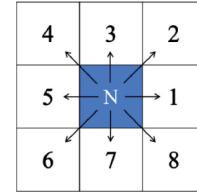


图 2 节点搜索图

Fig. 2 Node search map

表 1 搜索邻域关系表

Table 1 Search neighborhood relationship table

α	保留节点方向	舍弃节点方向
$[0^\circ, 45^\circ]$	1、2、3、4、7、8	5、6
$[45^\circ, 90^\circ]$	1、2、3、4、5、8	6、7
$[90^\circ, 135^\circ]$	1、2、3、4、5、6	7、8
$[135^\circ, 180^\circ]$	2、3、4、5、6、7	1、8
$[180^\circ, 225^\circ]$	3、4、5、6、7、8	1、2
$[225^\circ, 270^\circ]$	1、4、5、6、7、8	2、3
$[270^\circ, 315^\circ]$	1、2、5、6、7、8	3、4
$[315^\circ, 360^\circ]$	1、2、3、6、7、8	4、5

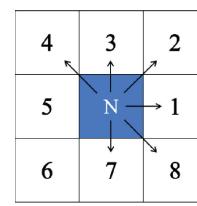


图 3 节点搜索优化图

Fig. 3 Node search optimization graph

2.4 去除冗余节点

A^* 算法规划出来的路径常常存在较多冗余节点,不利于移动机器人的运动,如图 4 所示。因此本文提出了一种根据安全距离去除冗余节点的策略,有效缩短了路径长度,提高计算效率和安全性。安全距离 d_s 设为一个栅格长度的 $5/6$ 倍,为障碍物中心到路径线段的距离。具体步骤如下:

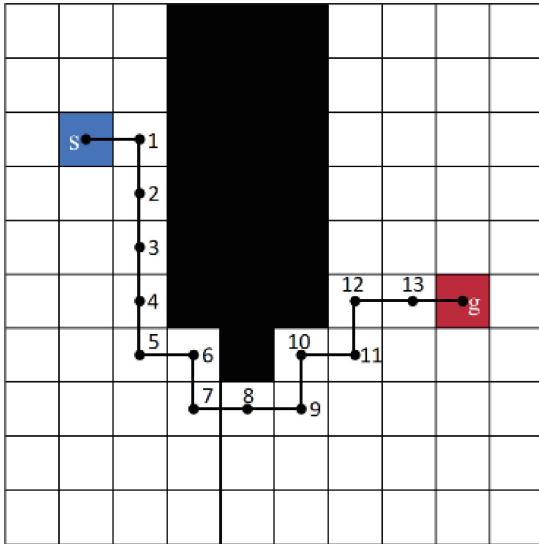


图 4 原始路径

Fig. 4 Original path

1) 遍历路径中的所有节点,将处于同一条直线上的中间节点删除,只保留起始点、拐点和终点,即 $s \rightarrow 1 \rightarrow 5 \rightarrow 6 \rightarrow 7 \rightarrow 9 \rightarrow 10 \rightarrow 11 \rightarrow 12 \rightarrow g$, 如图 4 所示。

2) 设现在的路径中节点为 $\{P_j \mid j = 1, 2, \dots, n\}$, 连接 $P_1 P_3$, 如果线段 $P_1 P_3$ 到障碍物中心的距离大于安全距离, 则继续连接 $P_1 P_4$, 直到线段 $P_1 P_j$ 到障碍物中心的距离小于安全距离或者穿过障碍物, 则保留节点 $j-1$ 和节点 j , 连接 $P_1 P_{j-1}$, 同时删除节点 1 和节点 $j-1$ 中间的节点; 然后从节点 j 开始重复上述步骤, 直至路径中没有冗余节点为止。新路径则为 $s \rightarrow 5 \rightarrow 7 \rightarrow 9 \rightarrow g$, 如图 5 所示。

2.5 路径平滑

经过节点优化后, 虽然提高了效率, 但是所规划的路径仍然存在着多个转折点, 导致路径不平滑问题。贝塞尔曲线是一种使用数学方式描述的曲线, 被广泛应用于绘图软件中以生成平滑的曲线^[23]。

一阶贝塞尔曲线为一条直线, 如图 6 所示, 在平面内任取两个点 P_0 和 P_1 , 连接线段 $P_0 P_1$, 在线段中间任取一点 A , 将线段 $P_0 A$ 和线段 $P_0 P_1$ 的比值设为 t , A 点的位置设为 $\{P(t) \mid t \in (0, 1)\}$, 就得到了一阶贝塞尔曲线公式:

$$P(t) = (1-t)P_0 + tP_1 \quad (10)$$

二阶贝塞尔曲线由平面内不共线的 3 个点 P_0, P_1 和 P_2 确定, 如图 7 所示, 连接线段 $P_0 P_1$ 和 $P_1 P_2$, 在线段 $P_0 P_1$ 上取一点 A , 线段 $P_1 P_2$ 上取一点 B , 连接线段 AB ,

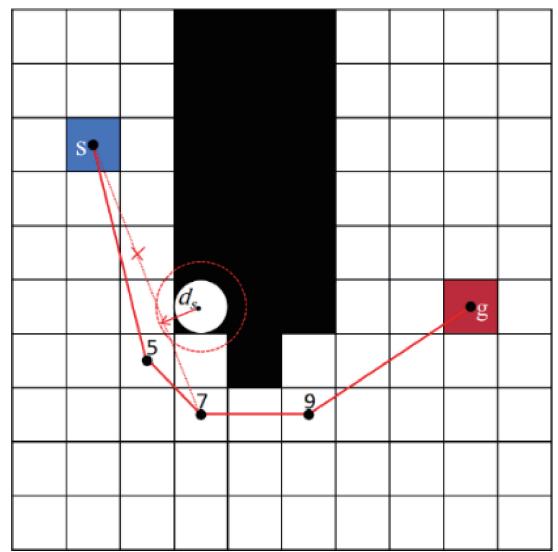


图 5 优化后的路径

Fig. 5 Optimized path

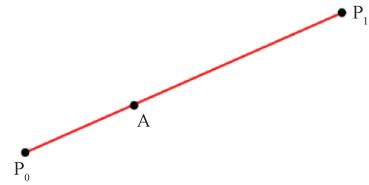


图 6 一阶贝塞尔曲线

Fig. 6 First-order Bézier curve

在线段 AB 上取一点 C , 这 3 点满足如式(11)所示的关系。

$$t = \frac{P_0 A}{P_0 P_1} = \frac{P_1 B}{P_1 P_2} = \frac{AC}{AB} \quad (11)$$

$$P_A(t) = (1-t)P_0 + tP_1 \quad (12)$$

$$P_B(t) = (1-t)P_1 + tP_2 \quad (13)$$

$$P(t) = P_C(t) = (1-t)P_A(t) + tP_B(t) \quad (14)$$

联立方程可得二阶贝塞尔曲线:

$$P(t) = (1-t)^2 P_0 + 2t(1-t)P_1 + t^2 P_2 \quad (15)$$

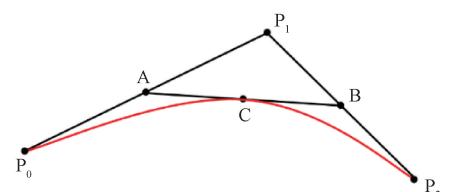


图 7 二阶贝塞尔曲线

Fig. 7 Second-order Bézier curve

由图 7 可知, 当 t 的取值逐渐从 0 增加到 1 时, A 点、 B 点和 C 点分别在各自的线段上从起点运动到终点, C 点的运动轨迹就是二阶贝塞尔曲线。

三阶贝塞尔曲线则由平面内 4 个点 P_0, P_1, P_2 和 P_3 确定, 如图 8 所示, A 点、 B 点、 C 点、 D 点、 E 点和 F 点满足如式(16)所示的关系:

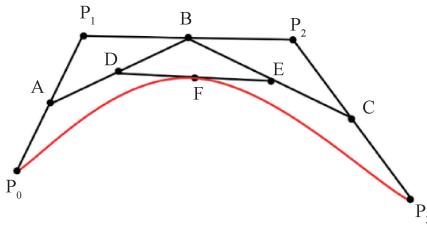


图 8 三阶贝塞尔曲线

Fig. 8 Third-order Bézier curve

$$t = \frac{P_0 A}{P_0 P_1} = \frac{P_1 B}{P_1 P_2} = \frac{P_2 C}{P_2 P_3} = \frac{AD}{AB} = \frac{BE}{BC} = \frac{DF}{DE} \quad (16)$$

其中, A 点、B 点和 C 点对应一阶贝塞尔曲线, D 点和 E 点对应二阶贝塞尔曲线, F 点的运动轨迹就是三阶贝塞尔曲线, 通过上述推导过程可得三阶贝塞尔曲线的公式如下:

$$P(t) = (1-t)^3 P_0 + 3t(1-t)^2 P_1 + 3t^2(1-t) P_2 + t^3 P_3 \quad (17)$$

由上述贝塞尔曲线的计算公式可得, 只有三阶贝塞尔曲线的二阶导数是连续的, 满足曲率连续变化的条件, 更加符合移动机器人的运动。故本文选用三阶贝塞尔曲线对路径进行平滑处理。

3 DWA 算法及其改进

DWA 算法的原理是将移动机器人的运动空间转化为速度空间 (v, ω) , 其中 v 和 ω 分别为机器人的线速度和角速度, 在这个速度空间中, 机器人的线速度和角速度被用来表示不同的速度组合, 通过对这些速度组合进行模拟, 预测移动机器人在一段时间内可能的运行轨迹, 然后根据预设的评价函数对每条轨迹进行评分, 最终选择得分最高的轨迹所对应的速度组作为移动机器人下一时刻的实际运行速度。

3.1 移动机器人运动学模型

通过建立移动机器人运动学模型, DWA 算法可以更好地模拟移动机器人的运行轨迹。设 t 时刻移动机器人线速度为 v_t , 角速度为 ω_t , 横摆角为 θ_t , 移动机器人运动学模型如图 9 所示, 数学表达式可以表示为:

$$\begin{cases} x(t+1) = x(t) + v_t \Delta t \cos \theta_t \\ y(t+1) = y(t) + v_t \Delta t \sin \theta_t \\ \theta_{t+1} = \theta_t + \omega_t \Delta t \end{cases} \quad (18)$$

3.2 速度采样

在移动机器人的速度空间 (v, ω) 内, 存在着无穷多个速度组, 但由于移动机器人自身的特性与所处环境的约束, 实际采样速度需限制在一定区间内。

1) 移动机器人自身速度约束:

$$v_s = \{(v, \omega) \mid v \in [v_{\min}, v_{\max}], \omega \in [\omega_{\min}, \omega_{\max}]\} \quad (19)$$

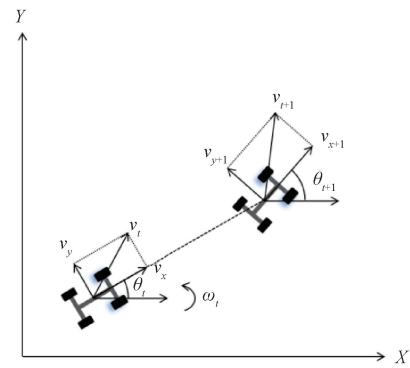


图 9 移动机器人运动学模型

Fig. 9 Mobile robot kinematics model

其中, v_{\min}, v_{\max} 分别为移动机器人最小线速度和最大线速度, $\omega_{\min}, \omega_{\max}$ 分别为移动机器人最小角速度和最大角速度。

2) 移动机器人电机加减速速度约束:

$$v_d = \{(v, \omega) \mid v \in [v_c - \dot{v}_b t, v_c + \dot{v}_a t], \omega \in [\omega_c - \dot{\omega}_b t, \omega_c + \dot{\omega}_a t]\} \quad (20)$$

其中, v_c 和 ω_c 分别为当前移动机器人的线速度和角速度, \dot{v}_a 和 \dot{v}_b 分别为移动机器人的最大线加速度和线减速度, $\dot{\omega}_a$ 和 $\dot{\omega}_b$ 分别为移动机器人的最大角加速度和角减速度。

3) 安全性约束:

在减速度为最大的条件下, 移动机器人必须在与障碍物发生碰撞之前将其速度降至零, 以确保整个运动过程的安全性。

$$\begin{aligned} v_a &= \{(v, \omega) \mid v < \sqrt{2 \text{dist}(v, \omega) \dot{v}_b}, \\ \omega &< \sqrt{2 \text{dist}(v, \omega) \dot{\omega}_b}\} \end{aligned} \quad (21)$$

其中, $\text{dist}(v, \omega)$ 为当前轨迹与最近障碍物的距离。

综上, 移动机器人的速度范围 v_r 表示为:

$$v_r = v_s \cap v_d \cap v_a \quad (22)$$

3.3 改进评价函数

对移动机器人的速度采样后, 根据评价函数对多组速度进行评分, 以选取最优轨迹。传统的评价函数公式为:

$$G(v, \omega) = \sigma [\alpha \cdot \text{head}(v, \omega) + \beta \cdot \text{dist}(v, \omega) + \lambda \cdot \text{vel}(v, \omega)] \quad (23)$$

其中, $\text{head}(v, \omega)$ 的计算公式为:

$$\text{head}(v, \omega) = 180^\circ - \theta \quad (24)$$

其中, θ 为移动机器人模拟的轨迹末端与目标点方向之间的夹角。

$\text{dist}(v, \omega)$ 为模拟轨迹与最近障碍物的距离, $\text{vel}(v, \omega)$ 为模拟轨迹对应的速度大小, σ 为平滑函数, α, β 和 λ 为各自代价子函数的加权系数。当 θ 越小, $\text{head}(v, \omega)$ 的得分越高; 当模拟轨迹离最近障碍物越远时, $\text{dist}(v, \omega)$ 的得分越高, 轨迹更安全; 当速度越大时, $\text{vel}(v, \omega)$ 的得分越高, 反之则评分越低。

由于传统的评价函数中速度代价子函数的系数为常量,常常因此陷入局部最优问题。希望离障碍物较远时,增加 $vel(v, \omega)$ 的占比,提高运行效率;离障碍物较近时,减少占比,安全躲避障碍物。为此本研究在 $vel(v, \omega)$ 添加了动态权重系数,同时在代价函数 $G(v, \omega)$ 中加入目标点代价子函数 $dist_g(v, \omega)$,减小行驶路径长度。改进后的 $G(v, \omega)$ 如下:

$$G(v, \omega) = \sigma[\alpha \cdot head(v, \omega) + \beta \cdot dist(v, \omega) + \lambda \cdot \ln(e + dist(v, \omega)) \cdot vel(v, \omega) + \gamma dist_g(v, \omega)] \quad (25)$$

其中, $dist_g(v, \omega)$ 的计算公式为:

$$dist_g(v, \omega) = \frac{1}{\sqrt{(x_g - x_m)^2 + (y_g - y_m)^2}} \quad (26)$$

其中, (x_m, y_m) 为轨迹末端的坐标, (x_g, y_g) 为目標点坐标, γ 为 $dist_g(v, \omega)$ 的加权系数。当轨迹末端距离目標点越小时, $dist_g(v, \omega)$ 的得分越高,反之则越低。

4 算法融合

通过将改进后的 A* 算法与 DWA 算法相融合,移动机器人可以实现全局最优路径规划,并能自主躲避动态障碍物。该算法首先使用改进 A* 算法进行全局路径规划,得到全局节点后,使用 DWA 算法对每两个相邻节点进行局部动态规划。具体算法实现如图 10 所示。

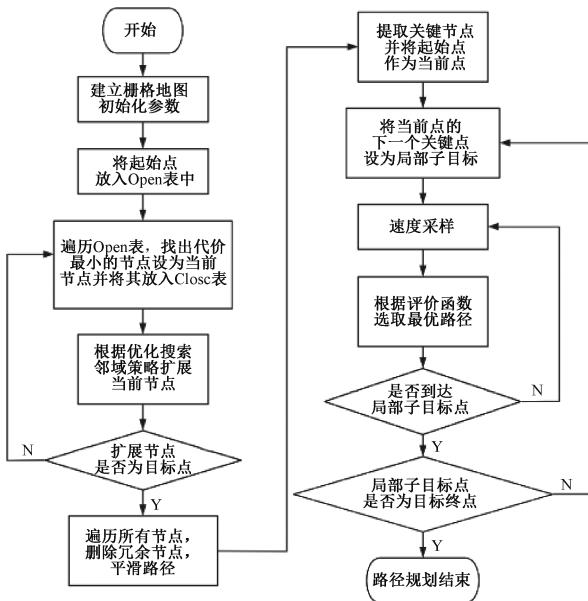


图 10 算法融合流程

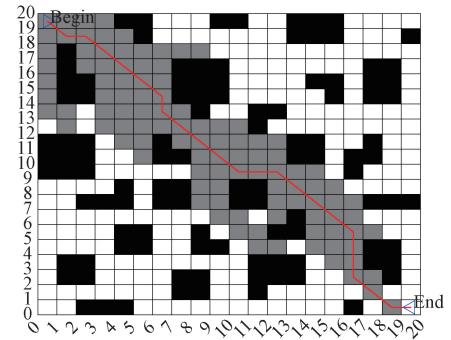
Fig. 10 Flow of fusion algorithm

5 仿真实验及结果分析

为验证所提出算法的有效性,本文分别在 20×20 和 30×30 两种尺寸的栅格地图进行仿真对比验证,运行环境为 Windows10 和 MATLAB R2022b。

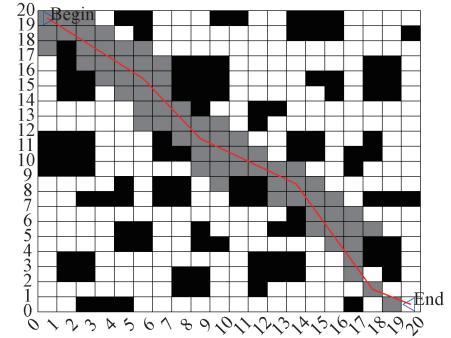
5.1 改进 A* 算法仿真实验分析

在环境地图相同的情况下,分别将传统 A* 算法、文献[24]与本文改进 A* 算法做对比,仿真结果如图 11 和 12 所示,其中灰色栅格表示为 A* 算法的搜索空间。环境 1 为 20×20 的栅格地图,起点坐标为 $(1, 20)$,终点坐标为 $(20, 1)$;环境 2 为 30×30 的栅格地图,起点坐标为 $(1, 30)$,终点坐标为 $(30, 1)$ 。



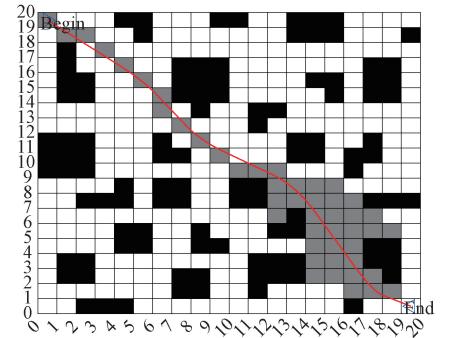
(a) 20×20 地图传统 A* 算法仿真结果

(a) Simulation result of conventional A* algorithm in 20×20 map



(b) 20×20 地图文献[24]改进算法仿真结果

(b) Simulation result of the improved algorithm of literature [24] in 20×20 map



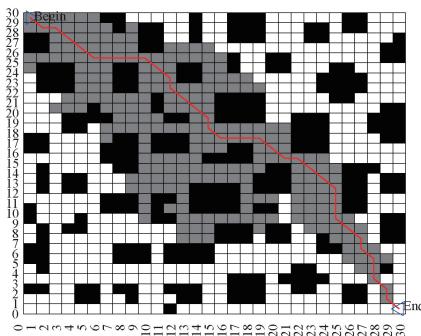
(c) 20×20 地图本文改进 A* 算法仿真结果

(c) Simulation result of the improved A* algorithm of this paper in 20×20 map

图 11 环境 1 中不同 A* 算法仿真结果

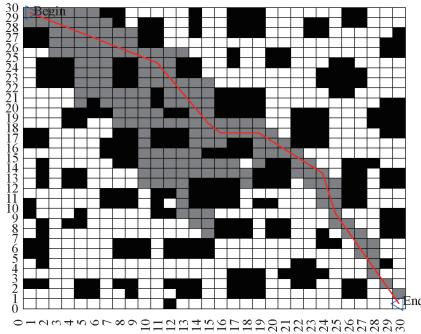
Fig. 11 Simulation results of different A* algorithms in environment 1

如图 11 和 12 所示,两种环境中传统 A* 算法、文献[24]改进算法与本文改进 A* 算法都能规划出一条全局路径。传统 A* 算法规划路径转折点过多,搜索效率低,文献[24]改进算法折线优化后规划路径与本文改进 A* 算法规划路径转折点都减少但搜索效率不如本文改进 A* 算



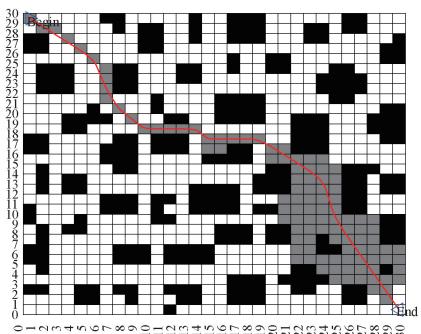
(a) 30×30地图传统A*算法仿真结果

(a) Simulation result of conventional A* algorithm in 30×30 map



(b) 30×30地图文献[24]改进算法仿真结果

(b) Simulation result of the improved algorithm of literature [24] in 30×30 map



(c) 30×30地图本文改进A*算法仿真结果

(c) Simulation result of the improved A* algorithm of this paper in 30×30 map

图 12 环境 2 中不同 A* 算法仿真结果

Fig. 12 Simulation results of different A* algorithms in environment 2

法。此外在环境复杂的情况下文献[24]改进算法规划的路径可能会与障碍物过于接近,而本文改进 A* 算法规划的路径不仅会与障碍物保持在安全距离之内,同时具有很好的平滑性,大大提高了移动机器人在运动过程中的安全性。

从表 2 中的数据可知,在环境 1 中本文改进 A* 算法规划的路径长度相比传统 A* 算法缩短了 6.36%,搜索节点数分别减少了 56.61%,规划时间减少了 28.34%;相比文献[24]改进算法路径长度减少 0.82%,搜索节点数分别减少了 32.84%,规划时间减少了 7.88%。在环境 2 中本文改进 A* 算法规划的路径长度相比传统 A* 算法缩短了 3.92%,搜索节点数分别减少了 57.48%,规划时间减少了 40.45%;相比文献[24]改进算法路径长度减少 1.17%,搜索节点数分别减少了 40.34%,规划时间减少了 9.09%。有效地证明了本文改进 A* 算法在提高安全性的同时也大大缩短了路径长度和规划时间。

5.2 融合算法仿真实验分析

为验证融合算法的有效性,在环境相同的情况下分别将传统 A* 算法融合传统 DWA 算法、文献[24]融合算法与本文改进融合算法做对比,并在地图中随机添加障碍物,以评估本文融合算法在动态环境下随机避障的能力。设置移动机器人最大速度为 1 m/s,最大加速度为 0.2 m/s²,速度分辨率为 0.01 m/s,最大角速度为 0.6 rad/s,最大角加速度为 0.24 rad/s²,角速度分辨率为 1 rad/s,评价函数权重系数 $\alpha = 0.3$, $\beta = 0.25$, $\lambda = 0.2$, $\gamma = 0.2$,预测窗口时间为 3 s。仿真结果如图 13~16 所示,其中虚线路径为 A* 算法规划路径,实线路径为融合算法规划路径。

如图 13~16 所示,传统融合算法虽然能够实时避开动态障碍物,但由于冗余节点过多导致规划路径过于曲折,路径长度与规划时间明显增加,降低了移动机器人行驶过程中的效率。文献[24]融合算法规划路径与本文改进融合算法规划路径的平滑性均优于传统融合算法,但本文改进融合算法规划路径长度和时间相比于文献[24]融合算法都有所减少,效率更高。

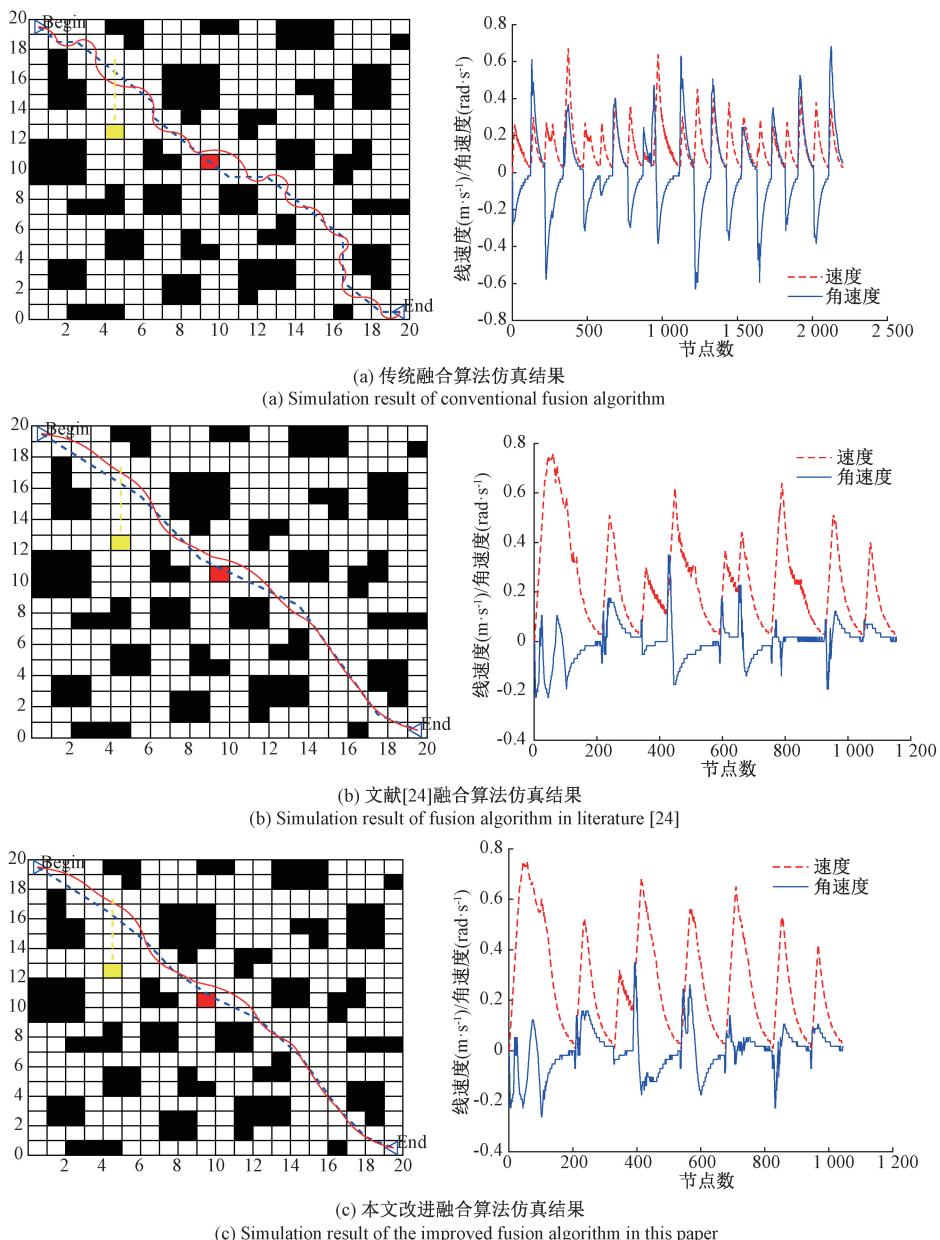
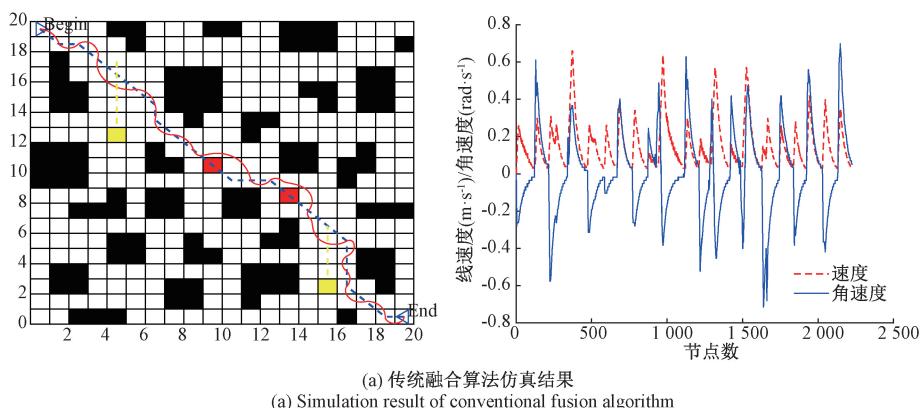
表 2 不同 A* 算法仿真结果对比

Table 2 Comparison of simulation results of different A* algorithms

算法	20×20			30×30		
	路径长度/m	搜索节点数	时间/s	路径长度/m	搜索节点数	时间/s
传统 A* 算法	29.213 2	106	0.063 33	46.284 3	247	0.094 86
文献[24]算法	27.583 4	67	0.049 26	44.997 5	176	0.062 14
本文改进算法	27.355 9	46	0.045 38	44.470 8	105	0.056 49

从表 3 中的数据可知,在 20×20 地图环境 1 中,本文改进融合算法规划的路径长度和时间相比于传统融合算法分别减少了 17.02% 和 52.61%,相比于文献[24]融合算法分别减少了 0.31% 和 9.6%;在 20×20 地图环境 2 中,

本文改进融合算法规划的路径长度和时间相比于传统融合算法分别减少了 17.26% 和 52.09%,相比于文献[24]融合算法分别减少了 0.72% 和 12.32%;在 30×30 地图环境 1 中,本文改进融合算法规划的路径长度和时间相比于

图 13 20×20 地图环境 1 中不同融合算法仿真结果Fig. 13 Simulation results of different fusion algorithms in 20×20 map environment 1

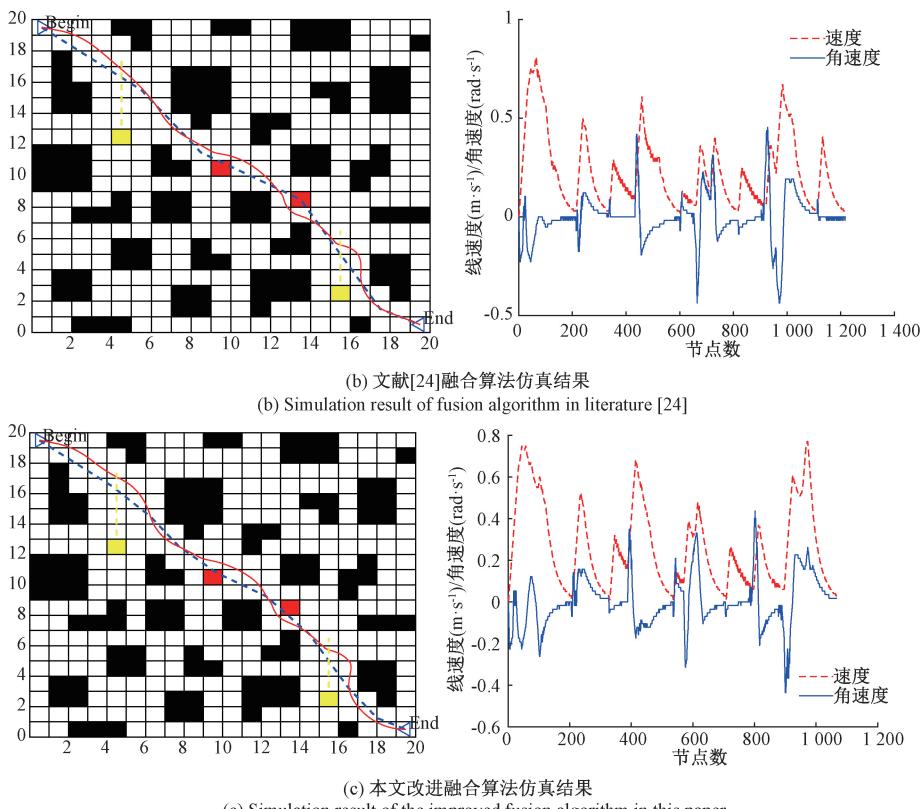
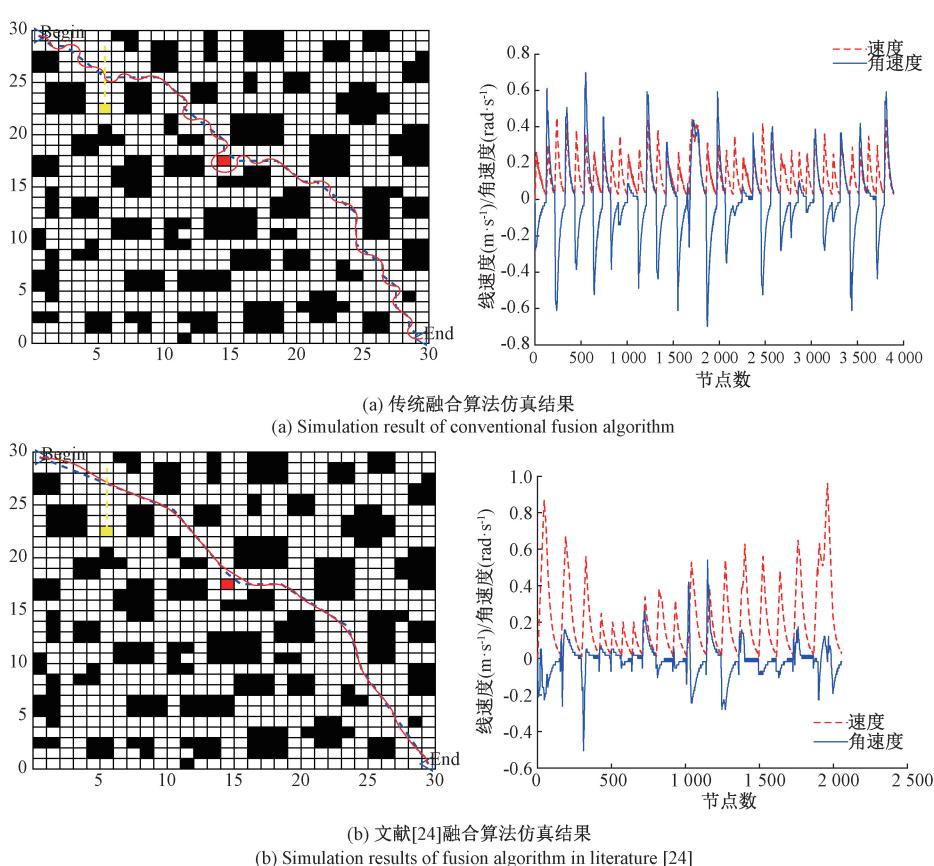
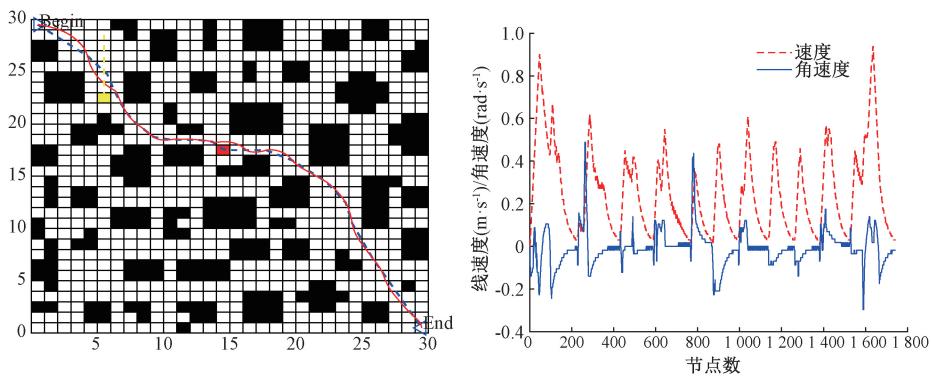


Fig. 14 Simulation results of different fusion algorithms in 20×20 map environment 2

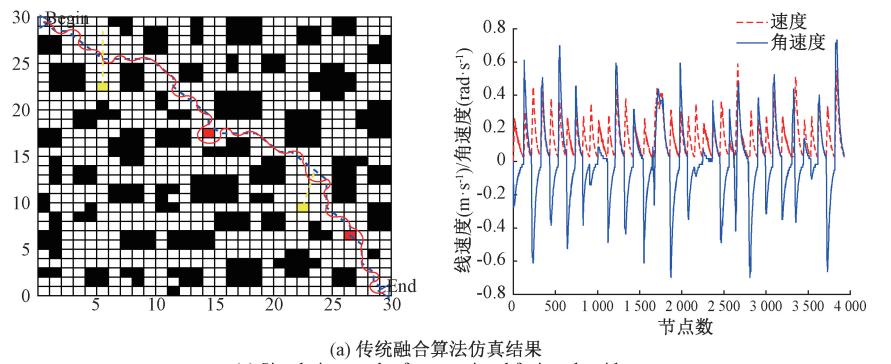




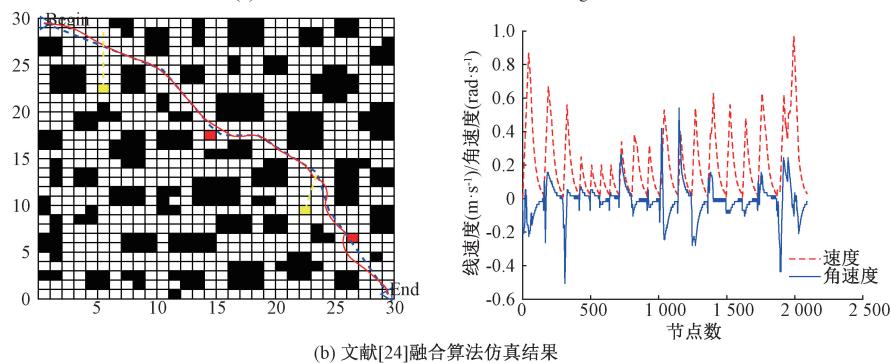
(c) 本文改进融合算法仿真结果
(c) Simulation result of the improved fusion algorithm in this paper

图 15 30×30 地图环境 1 中不同融合算法仿真结果

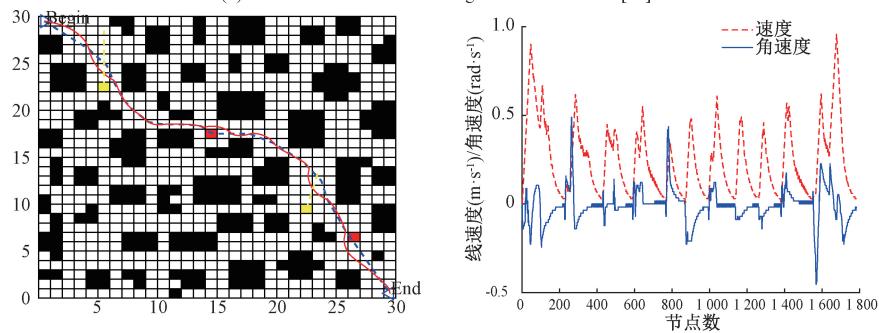
Fig. 15 Simulation results of different fusion algorithms in 30×30 map environment 1



(a) 传统融合算法仿真结果
(a) Simulation result of conventional fusion algorithm



(b) 文献[24]融合算法仿真结果
(b) Simulation result of fusion algorithm in literature [24]



(c) 本文改进融合算法仿真结果
(c) Simulation result of the improved fusion algorithm in this paper

图 16 30×30 地图环境 2 中不同融合算法仿真结果

Fig. 16 Simulation results of different fusion algorithms in 30×30 map environment 2

传统融合算法分别减少了 21.81% 和 55.28%，相比于文献[24]融合算法分别减少了 3.01% 和 15.35%；在 30×30 地图环境 2 中，本文改进融合算法规划的路径长度和时间相比于传统融合算法分别减少了 23.46% 和 54.64%，相比于文献[24]融合算法分别减少了 3.25% 和 14.76%。

表 3 不同融合算法仿真结果对比

Table 3 Comparison of simulation results of different fusion algorithms

环境信息	融合算法	路径长度/m	时间/s
地图环境 1 20×20	传统融合算法	33.643 9	220.5
	文献[24]融合算法	28.000 3	115.6
	本文改进融合算法	27.916 9	104.5
地图环境 2 20×20	传统融合算法	34.691 1	222.9
	文献[24]融合算法	28.911 6	121.8
	本文改进融合算法	28.704 5	106.8
地图环境 1 30×30	传统融合算法	55.939 2	389.8
	文献[24]融合算法	45.099 4	205.9
	本文改进融合算法	43.742 6	174.3
地图环境 2 30×30	传统融合算法	57.670 4	393.3
	文献[24]融合算法	45.626 5	209.3
	本文改进融合算法	44.143 4	178.4

综上看来，本文所提出的改进融合算法优于文献[24]所提出的融合算法，能够使移动机器人实现动态环境下的自主避障，符合预期标准。

6 结 论

本文针对传统 A* 算法在路径规划时存在搜索效率低、穿插障碍物、路径不平滑、无法规避未知障碍物等问题，提出了一种改进 A* 算法与改进 DWA 算法相融合的动态避障方法。在 A* 算法改进方面，通过引入全局障碍物占比率，在启发函数中增加动态权重系数，优化搜索领域，设定安全距离去除冗余节点，加入三阶贝塞尔曲线进行路径平滑。对于 DWA 算法的改进，在原有评价函数的基础上加入目标点代价子函数，同时动态调整代价函数的权重系数，最后将改进 A* 算法和改进 DWA 算法相融合，解决了以上问题。通过仿真实验对比验证，本文所提出的融合算法能够在复杂的动态环境下规划出一条最优路径，并实现实时避障，满足移动机器人行驶过程中的高效性和安全性。

参考文献

- [1] 崔炜, 朱发证. 机器人导航的路径规划算法研究综述[J]. 计算机工程与应用, 2023, 59(19): 10-20.
- CUI W, ZHU F ZH. A review of research on path planning algorithms for robot navigation[J]. Computer Engineering and Application, 2023, 59(19): 10-20.
- [2] WANG K, XU J, SONG K, et al. Informed anytime bi-directional fast marching tree for optimal motion planning in complex cluttered environments[J]. Expert Systems with Applications, 2023, 215: 119263.
- [3] AYAWLI B B K, CHELLALI R, APPIAH A Y, et al. An overview of nature-inspired, conventional, and hybrid methods of autonomous vehicle path planning[J]. Journal of Advanced Transportation, 2018, 2018(1): 8269698.
- [4] 翟文鹏, 刘润南, 朱承元. 基于改进 Dijkstra 算法的滑行路径优化[J]. 中国民航大学学报, 2022, 40(1): 1-6.
- ZHAI W P, LIU R N, ZHU CH Y. Glide path optimization based on improved Dijkstra's algorithm[J]. Journal of Civil Aviation University of China, 2022, 40(1): 1-6.
- [5] 孙岩霆, 王荣杰, 蒋德松. 融合 A* 与 DWA 算法的水面船艇动态路径规划[J]. 仪器仪表学报, 2024, 45(1): 301-310.
- SUN Y T, WANG R J, JIANG D S. Dynamic path planning for surface vessels by fusing A* and DWA algorithms [J]. Chinese Journal of Scientific Instrument, 2024, 45(1): 301-310.
- [6] 杨大庆, 谷侃锋. 基于快速扩展随机树的机械臂路径优化算法研究[J]. 组合机床与自动化加工技术, 2021(5): 10-12, 17.
- YANG D Q, GU K F. Research on robotic arm path optimization algorithm based on fast expanding random tree [J]. Combined Machine Tools and Automated Machining Technology, 2021 (5): 10-12, 17.
- [7] 何壮壮, 丁德锐. 基于 D-star 和 DWA 的改进机器人导航方法[J]. 电子测量技术, 2019, 42(12): 122-128.
- HE ZH ZH, DING D R. An improved robot navigation method based on D-star and DWA [J]. Electronic Measurement Technology, 2019, 42(12): 122-128.
- [8] 赵明, 郑泽宇, 么庆丰, 等. 基于改进人工势场法的移动机器人路径规划方法[J]. 计算机应用研究, 2020, 37(S2): 66-68, 72.
- ZHAO M, ZHENG Z Y, YAO Q F, et al. Mobile robot path planning method based on improved artificial potential field method [J]. Computer Application Research, 2020, 37(S2): 66-68, 72.
- [9] 严渐平, 黄俊儒, 吴迪. 基于 RRT 和 DWA 的欠驱动 UUV 路径规划[J]. 数字海洋与水下攻防, 2020, 3(3): 258-264.
- YAN ZH P, HUANG J R, WU D. Underdriven UUV path planning based on RRT and DWA [J]. Digital Ocean and Underwater Attack and Defense, 2020, 3(3): 258-264.
- [10] 唐嘉宁, 彭志祥, 李孟霜, 等. 基于改进 A* 算法的无人机路径规划研究[J]. 电子测量技术, 2023, 46(8): 99-104.
- TANG J N, PENG ZH X, LI M SH, et al. Research on UAV path planning based on improved A*

- algorithm[J]. Electronic Measurement Technology, 2023, 46(8): 99-104.
- [11] 张艳, 李炳华, 霍涛, 等. 融合改进 A* 算法与 DWA 算法的机器人动态避障方法研究[J/OL]. 系统仿真学报, 1-10 [2024-06-14]. <https://doi.org/10.16182/j.issn1004731x.joss.24-0143>.
- ZHANG Y, LI B H, HUO T, et al. Research on dynamic obstacle avoidance method for robots integrating improved A* algorithm and DWA algorithm [J/OL]. Journal of System Simulation, 1-10 [2024-06-14]. <https://doi.org/10.16182/j.issn1004731x.joss.24-0143>.
- [12] 赖荣燊, 窦磊, 巫志勇, 等. 融合改进 A* 算法和动态窗口法的移动机器人路径规划[J]. 系统仿真学报, 2024, 36(8): 1884-1894.
- LAI R SH, DOU L, WU ZH P, et al. Fusion of improved A* algorithm and dynamic window method for mobile robot path planning[J]. Journal of System Simulation, 2024, 36(8): 1884-1894.
- [13] 王文飞, 茹乐, 鲁博, 等. 基于改进 DWA 的无人机实时路径规划研究[J]. 电光与控制, 2023, 30(8): 50-55, 60.
- WANG W F, RU L, LU B, et al. Research on real-time path planning for UAVs based on improved DWA [J]. Electro-Optics and Control, 2023, 30(8): 50-55, 60.
- [14] 戴凯龙, 贾子彦, 潘玲俊. 基于改进动态窗口法的无人机避障航迹规划[J]. 智能计算机与应用, 2022, 12(7): 114-117.
- DAI K L, JIA Z Y, PAN L J. Obstacle avoidance trajectory planning for UAVs based on improved dynamic window method[J]. Intelligent Computer and Applications, 2022, 12(7): 114-117.
- [15] 王志特, 罗丽平, 廖义奎. 改进 A* 算法融合改进动态窗口法的移动机器人路径规划[J]. 计算机工程, 2024, 50(8): 86-101.
- WANG ZH T, LUO L P, LIAO Y K. Improved A* algorithm fused with improved dynamic window method for mobile robot path planning[J]. Computer Engineering, 2024, 50(8): 86-101.
- [16] 贺勇, 侯体成, 曾子望. 融合改进 A* 和动态窗口法的无人机路径规划[J/OL]. 机械科学与技术, 1-9 [2023-10-19]. <https://doi.org/10.13433/j.cnki.1003-8728.20230322>.
- HE Y, HOU T CH, ZENG Z W. Fusion of improved A* and dynamic window method for UAV path planning[J/OL]. Mechanical Science and Technology, 1-9 [2023-10-19]. <https://doi.org/10.13433/j.cnki.1003-8728.20230322>.
- [17] 邓云峰, 黄翼虎. 复杂动态环境下基于 A* 的改进 DWA 算法研究[J]. 电子测量技术, 2023, 46(9): 69-76.
- DENG Y ZH, HUANG Y H. Research on improved DWA algorithm based on A* in complex dynamic environment [J]. Electronic Measurement Technology, 2023, 46(9): 69-76.
- [18] 罗征志, 韩怡可, 张鑫, 等. 改进 RRT-Connect 与 DWA 算法的巡检机器人路径规划研究[J]. 计算机工程与应用, 2024, 60(15): 344-354.
- LUO ZH ZH, HAN Y K, ZHANG X, et al. Research on path planning of inspection robot with improved RRT-Connect and DWA algorithm [J]. Computer Engineering and Applications, 2024, 60(15): 344-354.
- [19] 贾千禧, 赵旭, 蒙志君. 基于改进 DWA 的动态境路径规划算法[J]. 科学技术与工程, 2024, 24(15): 6313-6319.
- JIA Q X, ZHAO X, MENG ZH J. A dynamic realm path planning algorithm based on improved DWA[J]. Science, Technology and Engineering, 2024, 24(15): 6313-6319.
- [20] 郭银景, 侯佳辰, 吴琪, 等. AUV 全局路径规划环境建模算法研究进展[J]. 舰船科学技术, 2021, 43(17): 12-18.
- GUO Y J, HOU J CH, WU Q, et al. Research progress of AUV global path planning environment modeling algorithm[J]. Ship Science and Technology, 2021, 43(17): 12-18.
- [21] MIN H, XIONG X, WANG P, et al. Autonomous driving path planning algorithm based on improved A* algorithm in unstructured environment[J]. Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering, 2021, 235(2-3): 513-526.
- [22] 迟旭, 李花, 费继友. 基于改进 A* 算法与动态窗口法融合的机器人随机避障方法研究[J]. 仪器仪表学报, 2021, 42(3): 132-140.
- CHI X, LI H, FEI J Y. Research on random obstacle avoidance method for robots based on the fusion of improved A* algorithm and dynamic window method[J]. Chinese Journal of Scientific Instrument, 2021, 42(3): 132-140.
- [23] 汤玉春, 王睿忠. 基于改进 A-star 与 DWA 相融合的移动机器人动态路径规划算法[J]. 智能计算机与应用, 2024, 14(2): 18-27.
- TANG Y CH, WANG R ZH. Dynamic path planning algorithm for mobile robots based on improved A-star fused with DWA [J]. Intelligent Computer and Applications, 2024, 14(2): 18-27.
- [24] 姜佩贺, 王敬, 桑忠启, 等. 改进 A* 与 DWA 的室内服务机器人路径规划研究[J]. 计算机工程与应用, 2024, 60(15): 327-335.
- JIANG P H, WANG J, SANG ZH Q, et al. A study on path planning for indoor service robots with improved A* and DWA [J]. Computer Engineering and Applications, 2024, 60(15): 327-335.

作者简介

鲁志, 硕士研究生, 主要研究方向为计算机视觉与机器人导航。

E-mail: 1799256427@qq.com

刘莹煌(通信作者), 博士, 讲师, 主要研究方向为机器人、传感器等。

E-mail: tonnnyala@126.com